Sādhanā Vol. 30, Parts 2 & 3, April/June 2005, pp. 141–158. © Printed in India

Trust management for e-transactions

VISHWAS PATIL and R K SHYAMASUNDAR

School of Technology and Computer Science, Tata Institute of Fundamental Research, Homi Bhabha Road, Colaba, Mumbai 400 005, India
e-mail:{vtp,shyam}@tifr.res.in

Abstract. There has been enormous increase in transactions and cooperative-computing services on the internet. This is both a technical and a social phenomenon. Transactions and services over the internet have global reach and users, known or unknown to the service provider, might be interested in availing access or participating in the cooperative transaction in a distributed manner. Thus, it is very important for service providers to identify and establish trustworthiness of potential collaborators, which they do by writing contracts (e.g. access control, security policies; the words contract and policy are used interchangeably) without violating the privacy and confidentiality laws that prevail across geographical boundaries. But as the system becomes complex and dynamic, contractual incompleteness arises since it becomes cumbersome to mention potentially large set of outcomes of the user's choice of action. Trust plays a crucial role in the design of optimal contracts; not all the relevant, valuable information on the user's choice of action is incorporated in the equilibrium contract. It may also be noted in that traditional transactions, the notion of *seeing is believing* plays a vital role. However, in e-transactions, this is not the case. The challenge is to see how in such a scenario trust can indeed be generated. Note that the presence of trust facilitates cooperative behaviour and allows for exchange to occur in situations where its absence would preclude trade.

In this paper, we shall present a comparative analysis of various approaches of trust management in practice that integrates technology with other factors. We shall also bring out the relative deficiencies and how these issues are tackled in our ongoing work that facilitates execution of optimal contracts.

Keywords. e-Commerce; security; trust management; access control; PKI.

1. Introduction

The utility and benefits of e-commerce are well known. Internet has become an important medium for disseminating information, doing commerce and business. The enabling technologies are helping the organizations to put their resources and work-flow systems etc. over the internet. The expansion of network access is driving an increase in interactions among people and between people and businesses (a set of interacting parties often called electronic community). It is widely acknowledged that e-commerce has fallen short of its expected potential in terms of applications. This can be attributed, among many factors, to the *lack*

of trust that participants have in e-commerce transactions. In real-world human supervised transactions, many transactions go through the notion of *seeing is believing*. However, such a notion is absent in the cliche of e-transactions. The absence of trust among unfamiliar people makes it necessary to rely on factors such as the party's philosophy (seeing is believing!), psychology and factors like transaction economics, risk, optimism etc., for making transactions. In the context of transactions over the Internet, it is not possible to use factors as mentioned above as such trust models do not exist. Before discussing the issues and approaches for trust management (TM) in e-commerce, let us look at the notion of trust.

Trust is the extent to which one party is willing to depend on somebody, or something, in a given situation with a feeling of relative security, even though negative consequences are possible. It is a pervasive notion and, as such, has been studied thoroughly in a variety of different fields, including the social sciences, economics and philosophy. An important observation from all these sources is that trust which in a sense is "*one individual's opinion of another*" is a subjective notion, and every individual decides whether to trust another based on the evidence available for personal evaluation (although one might delegate this decision to a more authoritative source in certain circumstances i.e. a notion of context is necessary). Also, trust is not symmetric: two individuals do not need to have similar levels of trust in each other. Even if two entities get the same evidence, they might not necessarily interpret this information in the same way. It is self-preserving and self-amplifying, and increases through periodic successful interactions, and degrades through disuse or misuse.

In terms of economic theory, trust is an instrument for carrying forward the trade. Individuals write trade contract, which helps them as a legal document if some dispute among them arises in future. For example, imagine a buyer, who requires a certain item (or service) from a seller. Let us suppose that the exact nature of the contract is uncertain as the complete nature of the item is not yet realized. In an ideal world, the parties would write a contingent contract specifying exactly which are the deliverables in each state. However, if the number of states is very large, such a contract would be prohibitively expensive. In such situations, the parties will write an incomplete contract. When the state of nature is clear, they will renegotiate the contract, since at this stage they know what kind of goods are to be traded or each one's responsibilities and accountabilities are clear (Hart & Moore 1999). It must be pointed out that there can be a series of incomplete contracts before the final one is reached. In a sense, this leads to multi-stage contracts (that are sequential in nature) or hierarchical. In fact, success of one stage increases the trust for the next level. When the setup of trade involves complex information and trust, contractual incompleteness arises endogenously. In such situations, such incomplete contract is the only efficient way for the transaction to succeed and is referred to as equilibrium contract or optimal contract. Thus, any TMM should support at least specification of contracts that are used in practice traditionally if not more.

It is becoming increasingly important for the electronic community to have means and methods for tackling trust related issues for e-transactions. We feel that trust in e-commerce can be enhanced by the following factors.

(1) Social factors (invoke and establish trust): reputation, familiarity,
(2) legal system (enforce trust): law enforcement and judicial system,
(3) organizational and procedural factors (enable trust): banks and their rules, and
(4) technology (enable and enforce trust): cryptography, protocol standards, tools.

In this paper, we shall discuss technological issues involved in trust; other issues are out of the scope of this paper. It is important to note that in the technology involved in present day

transactions, it is indeed possible to have human intervention at all stages. In other words, some of the technology has been accepted socially to be trustworthy. The large question is to see how the integration of technology and human intervention enhances the trust of both. Some of the characteristic features of any transactions are as below.

(1) Identification and authentication,
(2) message confidentiality,
(3) message integrity,
(4) non-repudiation,
(5) transparent transaction process,
(6) traceability and accountability.

Thus, in an integrated form of the e-transactions if we can provide a measure of trust (measure may not be uniform – it is somewhat relative to those on which the community has accepted to be trustful) on the above characteristics, it is indeed possible to trust an unfamiliar entity over the Internet for commercial transactions.

The first four characteristics can be directly provided using public-key infrastructures (PKI) framework like PGP (Zimmermann 1995), X.509 (Ford & Baum 2002), SPKI/SDSI (Clarke *et al* 2001) that have generated trust in the society. Access control models based on PKIs, like RBAC (Ferraiolo *et al* 2000), PolicyMaker (Blaze *et al* 1998), KeyNote (Blaze *et al* 1999) provide a specification language that allows users to specify the fifth characteristic involving organizational and procedural factors. Traceability and accountability issues are prominent when users come across unfamiliar, autonomous and mobile entities, i.e. entities not accountable to some mutually acceptable agency[1]. PolicyMaker, KeyNote allow users to write assertion-based rules for concluding traceability and accountability of communicating partner before initiating the transaction in case of insufficient trust. However, these frameworks use X.509 as underlying cryptographic framework and inherit X.509's disadvantages due to its top-down (centralized) architecture. SPKI/SDSI, a distributed PKI framework, is another powerful candidate but does not have a specification language to capture the subjective and non-symmetric nature of trust.

The underlying PKI framework plays an important role in the scalability of TMM. In centralized models like X.509, trust accumulates at the global CA (root certification authority) and trustworthiness of CAs directly matters to the trust relation between the relying entities. TMMs based on hierarchical PKIs fall short of establishing a provable trust link between its user and an unknown user from a totally different administrative domain having no cross-certification path established with the hierarchy. Also, the disadvantages of PKIs like stale information in certificates, CRLs (certificate revocation lists), certificate's inability to capture dynamic information, etc., become disadvantages of the TMM. TMMs based on on-line capability model (Miller *et al* 2001) addresses some of these shortcomings with additional cost. Several other TMMs have been proposed for unenforced transactions, where traceability and accountability is less, in distributed environment. PeerTrust (Xiong & Liu 2003), Poblano (Chen & Yeager 2003), EigenTrust (Kamvar *et al* 2003), Advogato (Levien & Aiken 2005), Appleseed (Ziegler & Lausen 2004), PageRank (Page *et al* 1998) are some of the reputation-based TMMs.

The rest of the paper is organized as follows: A comparative analysis of existing TMMs is given in §2. Evolution of trust and other non-technological parameters involved in trust

[1]Note that these features exist even in traditional commerce

formation are discussed in §3. In §4, we present our approach towards designing a generalized form of access control framework called *flexi*-ACL (Patil & Shyamasundar 2003a), that has provisions for hybrid types of access control mechanisms, which also permits specifications of alternative or a priori anticipated emergency measures of access control for services. The paper concludes in §5.

2. Comparative study of existing trust models

In this section, we shall provide a comparative analysis of various frameworks in practice as TM systems. PKI frameworks have been extensively used for realizing TMMs since they provide verifiable methods for executing contracts. We shall also see how the PKI-enabled TMMs are more expressive to handle complex contracts.

2.1 *PKIs as trust management systems*

PKIs provide a systematic framework to generate, distribute, and maintain the cryptographic key pairs required for achieving properties like authentication, authorization, data (communication) confidentiality, data integrity, non-repudiation of communications over the internet. These frameworks use a data-structure, called digital certificate, to populate user's cryptographic key and other credentials. The methods to distribute and maintain the digital certificates may be different in each of these frameworks. This is a very useful framework for trust establishment, enhancement and enforcement.

In this sub-section, we shall discuss three prominent PKI frameworks differentiated by their architecture, thus providing three distinct trust models.

2.1a *Pretty good privacy (PGP):* PGP was mainly envisaged for secure e-mail communication. In fact, even now, for this purpose, it has generated a good amount of trust. Of course, it has drawbacks but the trust is for class of applications or transactions. PGP is a zero-configuration model, i.e. without any pre-configured infrastructural setup, users can participate in it. Users of PGP generate an asymmetric key pair on their own and distribute their public-key through floppies or newspapers (or any other off-line medium), so that others can communicate securely with them. Users maintain public keys of other users in a local database, called keyring. While adding a new user and corresponding public key to the keyring, the owner of the keyring also assigns a non-negative trust value to that entry; emphasizing its trust over that binding. It is obvious that keys exchanged personally will get higher trust values, unlike the keys retrieved using the internet as an exchange medium.

Users of PGP may retrieve public-keys of interest from keyrings of other users, which have high trust values in the local database. Thus, users of PGP form a decentralized database of user's name-key bindings associated with a non-negative trust value. Users independently update their keyrings and may share it with others. This mesh of name-key bindings formed by closely connected users is called "Web-of-Trust". The users of this model run the risk of being manipulated by wrong or stale information provided by the members of "Web-of-Trust". Otherwise, this framework can provide verifiable means to do authentication, can support message confidentiality, integrity, non-repudiation of transactions but lacks accountability, and cannot have transparent processes because there is no authority which can vouch for principals when there is no or less trust between communicating peers. So, it cannot be used in its original form for e-transactions.

2.1b *X.509:* This framework has become a *de facto* standard for extracting cryptographic features required for e-transactions. Initial configuration involves a certification authority (CA), a trusted authority, accountable for issuance and revocation of digital certificates. Having the public-key of CA, users of this framework are ensured of the ability to correctly retrieve public-keys of others. Cross-certification among the CAs greatly increases user's ability to authenticate other users lying under external CAs. But the scheme also suffers from inability to ensure the freshness of user's public-keys and eventually the credentials associated with them.

X.509 was initially developed to authenticate users of X.500 global directory structure so that they can update their portion of the directory structure. Later on, the framework was extended to cater to various applications by embedding application-specific credentials into the certificate to achieve authorization, data confidentiality, integrity, and non-repudiation apart from authentication. Also, clubbing together all the user's authorizations into a single certificate, besides the name binding, is a serious privacy issue.

In X.509, authorizations can only flow from a CA to the subjects of the certificate and thus, forms a rigid structure (hierarchical or centralized). Further, authentication among users involves inescapable implicit trust on the intermediate CAs, which is not an acceptable norm for e-transactions or distributed applications. But this framework provides all the features that form the basis of trust formation.

2.1c *SPKI/SDSI:* This framework provides the authentication and authorization mechanism in a distributed environment. Instead of a CA determining the name and authorization binding, principals of the system are allowed to issue certificates. It is a bottom-up approach of devising a PKI, in which small autonomous islands of trusted principals can form a scalable distributed PKI by issuing certificates binding to principals in other autonomous islands. Unlike X.509, the principals can independently manage their local name spaces and can refer to names defined in other principals' name space. Name certificates are purely used for binding relationship between issuer of the certificate and the subject of the certificate. The authorization certificates deal with delegation of a particular authorization mentioned in the certificate from its issuer to the subject or a group of subjects specified in it. While issuing an authorization certificate the issuer also specifies whether the subject of the certificate can further delegate the authorization or not by keeping a flag on or off. In this framework, the onus of authentication and proof of authorization to do something is left to the user. On the user's part, the compilation of authorization proof is a highly distributed process in terms of finding group memberships and appropriate missing authorization links from various principals that might help in forming a chain (sometimes more than one) of certificates implicating the authorization proof. The user can submit one of the certificate chains (possibly the one that reveals only the necessary information for the transaction to succeed (Patil & Shyamasundar 2003b) to the service provider as a proof of policy conformance. Thus, though such a security framework gives its users much flexibility, it suffers from problems that are inherent to all PKIs (arising due to the static nature of digital certificates).

There is nothing like the possibility of withdrawing/immediate revocation of authorization in this framework. Therefore, authorizations are conferred to the subjects based on the trust levels at the time of issuance and it remains statically valid until the certificate expires. Once a principal issues an authorization certificate, it has no control over the future behavior of the subject of that authorization. The subject might lend over its authorization to other unauthorized entities or even further delegate (if the delegation is allowed) that authorization without conforming to the policy or contract under which it has acquired the authorization, breaching the trust of the issuer. Through an intelligent use of extended names, groups and

Table 1. Comparison between PGP, X.509, and SPKI/SDSI.

	PGP	X.509	SPKI/SDSI
Architecture	Bottom-up	Top-down	Bottom-up
Certificates	Only name certificates	Single certificate contains name and authorization info	Separate name and authorization certificates
Name space	Global (email)	Global	Local, extendable
CA characteristics	Web-of-Trust	Hierarchical	Each public-key can act as a CA
Authorization delegation	Not available	Strictly from CA to subjects	Arbitrarily among users, optional k-of-n subjects
Operational environment	Decentralized	Centralized	Decentralized
Suitability as a trust model	Poor	Acceptable	Acceptable, more expressive than X.509

threshold authorizations, it is possible to keep a check on the behavior of principals; however, it becomes cumbersome in highly distributed environments and it will not be always possible for a service administrator to write access policies or contracts ex ante, for avoiding the unwanted scenarios. In other words, such schemes are not scalable. It may further be noted that in a distributed system, the policies of the different administrative domains may change dynamically and independently from other collaborating domains. So, even making thoughtful use of the extended names, group definitions and threshold authorizations without a priori knowledge of user's behavior, may not necessarily help in writing ACLs or optimal contracts purely using the SPKI/SDSI framework. Table 1 provides a comparative view of the above discussed PKIs with respect to some of the TMM requirements.

2.2 *PKI-enabled trust management systems*

It is evident that TMMs purely based on PKI frameworks are good for simple setups where contract terms are clearly defined (complete contracts). In complex setups (possibly distributed), these models fail to capture the future behavior or outcomes arising due to the actions taken by entities involved in the setup. Researchers have tried to capture such policies by proposing frameworks like PolicyMaker (Blaze *et al* 1998), KeyNote (Blaze *et al* 1999) etc. An overview of these schemes is given below.

2.2a *PolicyMaker:* PolicyMaker is an unified approach for specifying and interpreting security policies, credentials, and relationships that allow direct authorization of security-critical actions. It is a language/tool in the development of services whose main goal is privacy and authenticity. It is an attempt to solve the need to find a suitably trustworthy copy of the public-key of someone with whom one wants to communicate. The general principles of this scheme are:

- unified mechanism: provides a common language for policies, credentials, and relationships,
- flexibility: succinctness, the framework accommodates the PGP and X.509 certificates with trivial modifications,

- locality of control: avoids the monolithic hierarchy of certifying authorities,
- separation of mechanism from policy: credentials-verification mechanism does not depend on credentials themselves or the semantics of the applications that use them.

PolicyMaker's ability to express security credentials and policies without requiring the application to manage a mapping between personal identity and authority is especially convenient in systems that include anonymity as a security requirement. It also provides ways to express the conditions under which an individual or an authority is trusted and the conditions under which trust may be deferred.

2.2b *KeyNote:* It is a notation for specifying local security policies and security credentials that can be sent over an untrusted network. It is similar in spirit to the approach of PolicyMaker. KeyNote accepts as input a set of local policy assertions, a collection of credential assertions, and a collection of attributes (action environment) that describes a proposed trusted action associated with a set of public-keys. Applying assertion predicates to the environment, it decides consistency of actions with local policy.

This scheme is monotonic; adding an assertion to a query can never result in a query having a lower compliance value that it would have had without the assertion and removing an assertion never results in increasing the compliance value returned by KeyNote for a given query. This property can simplify the design and analysis of complex network-based security protocols required for e-transactions. KeyNote and PolicyMaker use X.509 as underlying security infrastructure and are capable of writing more complex security policies than that are possible under X.509 alone.

2.2c *REFEREE:* REFEREE (Chu *et al* 1997) provides both a general policy-evaluation mechanism for Web clients and servers and a language for specifying trust policies. It places all trust decisions under explicit policy control; in the REFEREE model, every action, including evaluation of compliance with policy, happens under the control of some policy. That is, REFEREE is a system for writing policies about policies, as well as policies about cryptographic keys, PICS label bureaus, certification authorities, trust delegation, etc.

2.2d *Antigone:* The Antigone system (McDaniel *et al* 1999) is a middle-ware layer that provides flexible interfaces for defining policy in group applications. Central to the design of Antigone is the definition of a suite of mechanisms that provide the core secure group services. Through the composition of the Antigone mechanisms (and their underlying micro-protocols), an application may precisely define the group policy. The component design achieves technology and infrastructure independence; new mechanisms may integrate desired functionality without affecting other system services.

All these schemes are deficient in handling advanced security issues like; immediate revocation of authority, tracking dynamic changes pertaining to individual entities, containing flow of delegated authorizations etc., in distributed setups. Inability of these schemes to handle such issues is due to the off-line nature of underlying PKI infrastructure and the static nature of their basic building block; digital certificate (Patil & Shyamasundar 2004a). Ad hoc extensions to PKI frameworks have been done to induce online validation of certificates – OCSP (Ford & Baum 2002), but operating such a system is very costly and may not suitably cater to all ranges of applications over the Internet.

A typical application scenario under PolicyMaker/KeyNote and other script-based frameworks discussed above could be; access will be granted to a principal if the principal sends a signed request using his e-mail account. The compliance checking routine will first check the

requester's signature on the access request made along with appropriate authorization credentials and will also verify the e-mail header field "`From:`" appearing in the e-mail request.

2.3 *Capability-based trust management systems*

Recently, capability-based access control approaches traditionally used in operating systems have been adapted for applications in distributed environments. Access control system based totally on capability model is an on-line system as opposed to the off-line feature of PKI model. In this section, we shall discuss two such frameworks; E (Millet *et al* 2001; Miller & Shapiro 2003) and RBAC (Ferraiolo *et al* 2000).

2.3a *Language E:* E (Millet *et al* 2001; Miller & Shapiro 2003) is a simple, secure, distributed, pure-object, persistent programming language. It blends lambda calculus, security as capabilities, and modern cryptography. It provides cryptographic capabilities as an abstraction allowing economy of engineering effort in creating smart contracts. A capability is a pairing of a designated process with a set of services that the process provides. And when different capabilities make different behaviours from the same process, we can view the process as a composite and each of its capabilities as a facet. These processes can cooperate-operate without vulnerability because all authorities are accessed only by references. Therefore, the credential proofs are the outcome of the relationships between the processes/entities at that particular time. This way the language E plans to tackle the complex relationships between the entities in a distributed environment by confining the delegated authorities.

2.3b *RBAC:* Role-based access control (Ferraiolo *et al* 2000) is a framework designed for simple management of permissions by associating them with roles, and users with appropriate roles. The roles can be granted new permissions as new applications and systems are incorporated, and the permissions can be revoked from roles as need arises. This approach is more stable because an organization's activities or functions usually change less frequently in contrast to the user's membership to roles. RBAC is policy-neutral by itself. It allows to express hierarchical structure of roles, possibly with constraints. Extensions to RBAC also allow us to express access policies spanning across more than one administrative domain.

Though these schemes support the basic characteristics required for trust formation and enforcement and provide mechanisms to handle complex issues like immediate revocation, effective confinement of delegated authority, they fail to write policies for engagement with unknown principals or electronic communities over the internet. Online schemes are generally good for setups involving well-defined contracts (complete contracts).

Engaging unfamiliar entities into the transaction of common interest or collaboration is a necessary requirement for doing trade over the Internet. Researchers have proposed several schemes for such scenarios, based on reputation of unknown entities. Several applications exist that have direct analogy with this scenario; e.g., P2P computing, mobile ad hoc networks etc. In next section, we shall discuss reputation-based schemes in practice.

2.4 *Reputation-based trust management systems*

On a decentralized network (zero configuration situation - ad hoc network), like peer-to-peer (P2P), users can see from where the information arrives, as well communicate their opinions on both the information they have acquired, and the peers who are its source. These personal opinions can be collected, exchanged, and evaluated. Furthermore, these opinions, when evaluated, can be used as guidelines for searching for information, and recommending information sources, thus, creating decentralized, personalized "Web-of-Trust". Trust becomes a

social contract with social implications for the participants. Each user of such decentralized setup will develop a reputation among his peers. Let us see few prominent reputation-based TM schemes.

2.4a *PeerTrust:* PeerTrust (Xiong & Liu 2003) aims to develop a trust mechanism for P2P networks so peers can quantify and compare the trustworthiness of other peers and perform trusted interactions based on their past interaction histories without trusted third parties. This model has two main features. First, it has three basic trust parameters and two adaptive factors in computing trustworthiness of peers, namely, feedback a peer receives from other peers, the total number of transactions a peer performs, the credibility of the feedback sources, transaction context factor and the community context factor. Second, it defines a general trust metric to combine these parameters.

2.4b *Poblano:* Poblano (Chen & Yeager 2003) is a JXTA-based effort to establish a reputation based decentralized trust model. It provides protocols for disseminating trust and algorithms required to update trust. This model is useful in performing reputation guided search and designing a recommendation system for security purposes. It tries to emulate PGP.

2.4c *EigenTrust:* EigenTrust (Kamvar *et al* 2003) in its basic form, is intended to decrease the number of downloads of non-authentic files in a P2P file-sharing network using a reputation model. In this scheme, each peer is assigned a unique global trust value, based on the peer's history of uploads. The algorithm provided in this scheme allows a user to compute global trust values in the distributed environment. By having peers use these global trust values to choose the peers from whom they download, the network effectively identifies malicious peers and isolates them from the network.

2.4d *Advogato:* Advogato (Levien & Aiken 0000) provides a trust metric, called Advogato maximum flow trust metric, in order to discover which users are trusted by members of an on-line community and which are not. Hereby, trust is computed by a centralized community server and considered relative to a seed of users enjoying supreme trust. However, the metric is not only applicable to community servers, but also arbitrary agents which may compute own trusted peers and not for the whole community they belong to. In this case, the agent itself constitutes the singleton trust seed. This model does not provide support for weighted trust relationships.

2.4e *Appleseed and PageRank:* Appleseed (Ziegler & Lausen 2004) can be seen as an improvement over Advogato model in terms of computation of local trust metric and its propagation. It also has working similarity with Google's PageRank (Page *et al* 1998) ranking algorithm.

A typical application scenario for above mentioned reputation-based framework can be described as follows: a principal would like to interact with a peer, for certain download, who is serving other peers in the setup for a long time with a satisfactory quality of service and against whom it has not heard of any complaint from the community. It may also consider its own past experience with that peer while making a decision on current requirement. But it may still get cheated by downloading a virus instead of an intended code. This is the risk that the relying party is taking even though there is a perceived risk associated with the transaction.

2.4f *TM systems based on rapport, introduction: Resurrecting duckling:* Usually, authorization mechanisms involve a centralized system administrator and that may be implemented

as access control lists or capabilities. But, in the zero-configuration environment the central authority will be absent and one has to establish the secure transient association between the entities coming into the environment. This is done under this framework by *imprinting*. As soon as an entity is exposed to intended environment, the entity will recognize as its owner the first entity that sends it a secret key and will remain faithful to its owner (Stajano & Anderson 1999). Thus, there will be a trusted environment in which all entities are faithful to their owner (controller). Having such trusted islands one can scale it up to larger community.

Apart from parameters like reputation etc., while engaging with unfamiliar entities over the Internet, various other non-technological parameters are also involved whose value or intensity of effect on individual's decision making is subjective. In next section, we shall discuss such parameters that have been extensively studied in other branches of science and have strong relevance for e-transactions.

3. Role of non-technological parameters in trust formation

The concept of trust has been addressed within many disciplines, including philosophy, psychology, sociology, transaction economics, and organization theory. This has resulted in a body of research that is widely divergent and at times contradictory. It is widely acknowledged that trust is complex and multidimensional. However, research often focuses narrowly on specific aspects of trust, failing to fully capture its multidimensional nature. Following are the important factors that variably affect an individual's decision-making process before transacting with an unfamiliar entity over the internet.

- *References*: References greatly enhance trust between the unfamiliar communicating partners. Two potential collaborators, unfamiliar with each other, may look for a principal commonly known to both of them. In fact, it is always possible to find a common reference known to both collaborators, directly or indirectly. The reference is said to be direct when only one referee who knows both collaborators first hand, otherwise it is said to be indirect. Harvard Professor Stanley Milgram has suggested that we are separated from every one else in this world only by six others at the most (Milgram 1967; Kleinberg 2000). However, direct references generally fetch more trust value than the indirect reference but an individual's optimism, intention behind the collaboration also determines the trust value. References provide some sort of guarantee/non-repudiation/traceability when such things are not possible from underlying technological framework.
- *Optimism, intentions and risk*: Optimism is the positive expectation a principal has for another principal or an organization based on past performance and truthful guarantees. Trust is about expectations of the future. Also, a principal's intentions behind collaboration plays an important role in the formation of a successful collaboration. But the extent to which a principal can be optimistic is capped by the risk associated with the unsuccessful or undesired outcome of the transaction. Principals go ahead with transactions with unfamiliar entities even though they are aware of perceived risk, it is because of either their intentions behind the transaction or their belief in evidences associated with the transaction or both of these.
- *Belief and transitivity of trust*: Harbison and Christianson, in their position paper (Christianson & Harbison 1997) state that trust is an epistemic notion: statements about trust are statements about certain beliefs held by others and their reasons for holding them, not about what would make such beliefs true in the real world. The derivation of comfortable trust value for a communicating partner involves one's belief in the available

set of associated evidences, willingness to take risk, intentions, and his own set of contextual parameters. We should not expect users to behave monotonously on these basic parameters in trust formation. Therefore, a generic trust management framework should provide means and methods for the users to express their beliefs (possibly conditional) while avoiding unintentional transitivity of trust (Christianson & Harbison 1997). It is difficult to achieve a natural TMM by providing a monotonous rule set, as is the case in most of the existing frameworks used as TMMs.

Trust increases through periodic successful interactions, and it degrades through disuse or misuse. Based on the severity of perceived risk associated with applications, users enter into collaboration with one of the four distinguishable regions of following figure 1 as the initial trust value. For example, in transactions of high financial value, users might not trust the communicating peer easily and may initiate the negotiation with the potential collaborator by picking a value in first quadrant (near zero trust value). The trust between the collaborators may gradually increase after enough number of successful transactions and in future they might collaborate for transactions of higher value. Micro-payment schemes like *e-coupons* (Patil & Shyamasundar 2004b) allow users to specify the financial worth of the fundamental monetary unit, are necessary to implement TMMs. These features provide some capability to limit the risk involved in transacting with strangers and can aid in evolving towards macro-payment mode (by increasing the intrinsic value of fundamental monetary unit or efficiently paying more units) after gathering/building enough trust with the peer. We observed similar risk-management feature in the vendors of our micro-payment system *e-coupons*, where vendors take a risk of accepting already spent payment-values from users. Since the protocol is an off-line protocol, for the sake of efficiency and cost, vendors do not verify payments made by users at the very beginning of their engagement. Vendor may come to know about the attempts of double-spending only after it sends the collected payment information to bank. Upon finding the effort of double-spending, vendor will stop trading with such user. In fact, there are several micro-transaction systems that just work on randomized billing that works out to be okay on an average, assuming a certain scenario of usage (in a sense, various

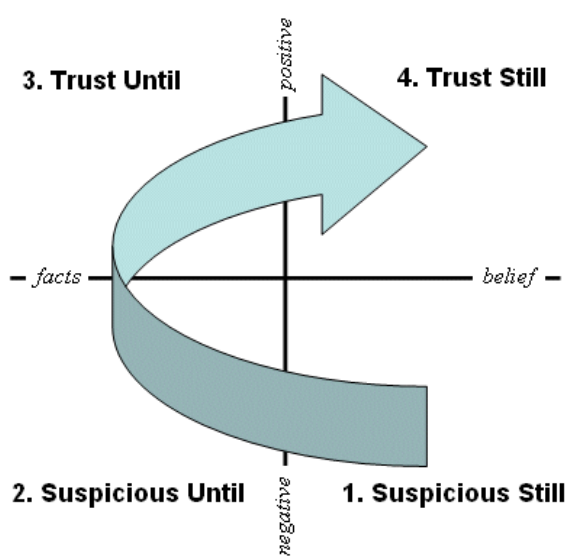

Figure 1. Evolution of trust (conceptualized by Paul English and Robert Fischer).

schemes for cell phone services also follow such schemes). In these scenarios the trust comes from the following aspects.

(1) The transaction value is low,
(2) the average billing meets the expectation of the user,
(3) double spending is not frequent, and
(4) the user can withdraw when he finds conditions (2)–(3) are not satisfied.

On the contrary, in applications like e-mails, principals let anyone send them mails, then start blocking the senders (spammers) who have breached the trust. In these situations, the principal offers the transactions with trust (assumes that parties do not send e-mails unnecessarily in a continuous way) and blacklists those who violate the accepted trust or the protocol.

The above scenarios indicate the complexity of trust even in such simple domains. Our study shows that none of the existing TMMs provide means and methods to write policies that takes into account all the factors in trust formation, enhancement and enforcement. Most of the TMMs are efficient when the setup is simple and small (scalability is a real issue). They work fine for *complete contracts*, but fail to handle *optimal contracts* or the complexity when the contracts are *incomplete*.

Economists show that an optimal response to complex information in agency relationships is for the contracting parties to choose exceedingly simple contracts – a fixed wage plus a discretionary bonus, for instance (Al-Najjar & Casadesus-Masanell 2001; Hart & Moore 1999). In the virtual world, we have technological means for enforcing simple complete contracts – fixed wage, but we do not have a comprehensive framework that fairly captures the incompleteness in contract induces by – discretionary bonus. *flexi*-ACL is an effort in this direction. This framework provides means and methods for its users to define the acceptable boundary conditions over collaborating peer's behavior while entering into incomplete contracts. Thus, allowing each individual entity of the system to clearly express the acceptable conditions under which it wants to collaborate with any other peer (possibly unknown). In the next section, we shall discuss the features captured in the *flexi*-ACL framework to execute complex, incomplete contracts.

A flexible and manageable trust management framework which can scale across the inter-organizational autonomous administrative domains, remains a key issue. Also the business topology demands an environment which is distributed and supporting methods to seamlessly interact between the distributed entities. Let us briefly enumerate the desired properties of a generic TMM.

(1) It should efficiently perform policy compliance check, against locally defined policy, over a set of credentials provided along with request and decide whether compliance with the local policy has been proven, and may also provide reasons for non-compliance along with additional information about how to proceed,
(2) should have provisions for envisaging risk management strategy, fault-tolerance under policy specification language,
(3) it should allow each entity to maintain his own local, statistical view of the whole trust network,
(4) should be scalable and manageable,
(5) should be expressive enough to granularly address policies affecting individual entity (since for the sake of manage-ability, one introduces abstraction of class of entities falling under a common security or trust policy),

(6) should be flexible enough to express policies spanning across different administrative domains,
(7) should allow users to import trust values computed by other trusted entities for unknown entities (i.e., specifying deferred policies based on policies defined elsewhere – extended policy specification),
(8) should be sound; A TMM is said to be *sound*, under which existing trust relationships can only be "qualified" and "combined" to form trust relationships with new characteristics.

By design, TMM unifies the notions of security policy, credentials, access control, and authorization. A generic TMM should support a variety of methods to an user to express his trust policy based on his belief, intentions, willingness to take risk and the context. The designer's goal is to engage users quickly, establish and preserve strategic trust under challenging situations. Requirements concerning security, reliability, and availability must be made by trading off costs and benefits and identifying acceptable levels of risk.

4. *flexi*-ACL

flexi-ACL (Patil & Shyamasundar 2003a; Patil & Shyamasundar 2004a) is an expressive access control policy specification framework in which, SPKI/SDSI is used as the underlying security infrastructure. This framework integrates *a priori* defined authentication mechanisms (on-line/off-line) with the underlying security infrastructure to perform complex and dynamic access policy conformance checks over incoming access requests. This framework is capable of serving advanced properties like immediate revocation of authorization (contract), confinement of delegated authorization, provisions for alternate authorization policy as an emergency measure etc. (Patil & Shyamasundar 2004a) apart from the basic characteristics like non-repudiation, confidentiality, traceability and accountability, required for on-line transactions in a secure and trusted manner. Therefore, *flexi*-ACL can be used as a powerful replacement for other traditional access control mechanisms (like pure PKIs as TMMs) as an underlying mechanism for constructing TMMs. The relation or hierarchy between different frameworks involved in the construction of a TMM is shown below in figure 2.

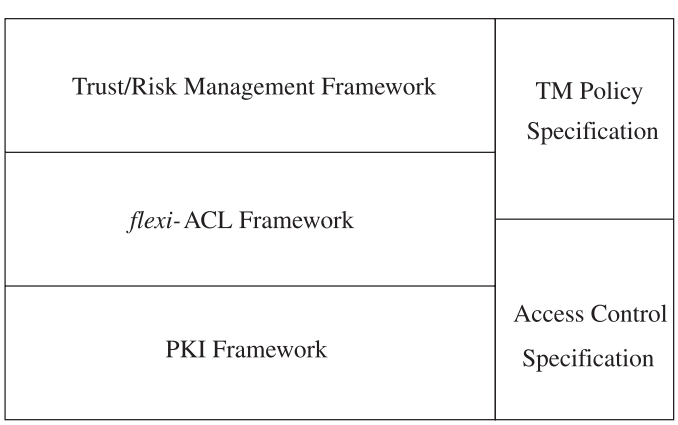

Figure 2. Hierarchy of frameworks in construction of a TMM.

```
(flexiacl
  (quorum 1
    (aclblock B1
      (rule X1) AND (rule X2) AND (rule X3))
    (aclblock B2
      (rule X1) AND ((rule Y1) OR (rule Y2)))))
```

Figure 3. *flexi*-ACL: Typical structure.

Readers are encouraged to refer (Patil & Shyamasundar 2003a; Patil & Shyamasundar 2004a) for complete understanding of *flexi*-ACL functionality, syntax and semantics. *flexi*-ACL segregates access control policy into global and local policy. By doing so, it empowers the resource controller (service provider) to implement properties like immediate authorization revocation of non-conforming users, fine-grained access to users, users' rights amplification etc., and brings modularity and manageability to the setup (Patil & Shyamasundar 2004a). The approach of segregating the more general access control policies, called global policy, from the local policies that are subjected to change more frequently or dynamically, helps us in realizing execution of incomplete contracts. The global policies are generally imbibed into the digital certificates since they are not subjected to frequent modifications and can be enforced or verified in an off-line and provable fashion. Whereas the local policies that are subjected to change based on context are maintained at the resource controller's side (Patil & Shyamasundar 2004a). Local policies are responsible for fairly enforcing the incomplete terms implicated into the contract (overall policy).

Figure 3 shows a typical access control policy designed using heterogeneous e-authentication mechanisms (like `spki`, `pamd`, `RSA SecurID`, `biometric`, `TCP/IP wrapper`, `token` etc.,) as basic access control `rules`[2]. The heterogeneous authentication mechanisms are glued together using the boolean `AND`, `OR` operators.

The heterogeneous primitive access control `rules` can be aggregated into an `acl-block`. Such blocks can be placed under another construct called `quorum` to achieve a very flexible access control policy expression. Such an unique, modular method of expressing access control policies allow a resource administrator to establish a comfortable trust or credibility level before making access control decisions. For the policy specified in figure 3, the requester has to conform to any of the two `acl-blocks` to get resource access, since the `quorum` is set to 1.

Let us analyse the sample *flexi*-ACL policy shown in figure 3, by assigning suitable types to the `(rule)` structures. For the sake of simplicity, we shall use only `spki`, `TCP/IP wrapper`, and `token` types for the analysis.

- `acl-block B1` :=
  - `(rule X1)` :=
    `(type := spki)`
    `(spki := my-collaborators (delegate)`
    `(tag (ftp://tifr.res.in (read write))))`
    Principals who can provide a certificate chain proving their membership to the group `my-collaborators` will evaluate this rule to `TRUE` or 1. Contracts of

[2]The complete details of *flexi*-ACL specifications are available in (Patil & Shyamasundar 2003a)

Table 2. `AND`, `OR` operators.

Expression	Description
`(rule X1) AND (rule X2)`	returns 1; when *evaluate* `(rule X1)` = 1 and *evaluate* `(rule X2)` = 1, else returns 0
`(rule Y3) OR (rule Y4)`	returns 0; when *evaluate* `(rule Y3)` = 0 and *evaluate* `(rule Y4)` = 0, else returns 1

this nature are used to distinguish potential collaborators on the basis of possession of credentials.

– `(rule X2)`:=
`(type := TCP/IP wrapper)`
`(TCP/IP wrapper := 158·144·0·0/16)`
To successfully evaluate this rule, principals should be making access request from an IP address belonging to the range specified in this rule. These rules are used to gather evidence about potential collaborators.

– `(rule X3)`:=
`(type := token)`
`(token := (issuer (some-notary.com)) (policy XYZ))`
To successfully evaluate this rule, principals should produce a cryptographic `token` issued by a principal trusted by the service provider. Service provider may specify a more trusted `issuer` for transactions of high risk. These kinds of rules are helpful while engaging unfamiliar collaborators based on the references they are able to provide. The unfamiliar entity's ability to obtain a `token` from the specified issuer is determined by his capability to satisfy policies defined at the token issuer's end (which can be of type *flexi*-ACL again!)

As we can see from figure 3 that the quorum is set to 1, a potential collaborator has to satisfy any of the `acl-block`s, i.e. either `acl-block B1` or `acl-block B2`. Furthermore, to satisfy the sub-policy specified in `acl-block B1`, one must evaluate the expression to {`TRUE`} (i.e. the return value should be equal to `(1)`) since the set of rules in this block are chained together by `AND` operator (see table 2). The rule, `(rule X1)`, is of type `spki`, based on SPKI/SDSI security infrastructure, and is helpful in enforcing complete contracts, for which the terms are dictated inside the static digital certificates. The auxiliary terms of incomplete contracts are achieved through `(rule X2)` and `(rule X3)`, based on the evidence associated with the transaction and successful vouchers made for the potential collaborator by the notary respectively.

- `acl-block B2`:=

  – `(rule X1)`:= `(type := spki)`
  `(spki := my-collaborators (delegate)`
  `(tag (ftp://tifr.res.in (read write))))`
  – `(rule Y1)`:=
  `(type := TCP/IP wrapper)`
  `(TCP/IP wrapper := 158·144·64·0/26)`
  In this rule, a shorter range of IP address from which principals should originate their access request is specified. In this fashion, service provider is demanding a more stricter or stronger evidence from the potential collaborators.

– `(rule Y2)`:=
`(type := token)`
`(token := (issuer (notary.gov)) (policy XYZ))`
In this rule, the service provider has specified a highly trusted notary from whom it is willing to accept `token`.

`acl-block B2` can be successfully evaluated if principals can successfully evaluate its constituent, i.e. `(rule X1)` – as described above and either of `(rule Y1)` or `(rule Y2)`. These rules are similar to `(rule X2)` and `(rule X3)` respectively, except that a strong evidence and a strong reference is demanded from the potential collaborators.

Therefore, any principal who is capable of generating a `spki` type of proof showing its membership to the group `my-collaborators` defined by the owner of ftp-server `ftp://tifr.res.in` can access/modify the digital content stored on the server, provided that it is used for research purpose and no commercial gain is made out of it. We broke this incomplete contract into static part (i.e. prove your membership) and auxiliary part (i.e. you are using the service for research purpose). IP range of `158·144·0·0/16` is substantially allotted to .res.in organizations (research organizations), so the access requests should originate from these IP addresses. However, `acl-block B2` provides some relaxation on this requirement, a `token` from a trusted server will suffice for obtaining comfortable trust value to go ahead with the transaction. In this fashion, *flexi*-ACL provides means and methods for users to express the trust relationships in a manageable and natural way.

From the above highlights, the reader can see that *flexi*-ACL supports specification of hierarchical or multi-stage contracts that require the notion of state. Further details can be obtained from the papers referred to.

Integration of a TMM with an application shall run from initial phase of finding trustworthiness of the peer (at the same stage where a PKI-based framework provides authentication for known peers) till the completion of the transaction by recording the transaction experience for future use.

5. Conclusions

Trust in information services and technologies has become an increasingly important issue for today's ever-changing networked world. The development of trust among business, consumers, and other stake-holders is seen as crucial to the expansion of e-business and the full exploitation of technological developments in this area. However, the way in which trust may be gained in this context is not yet well understood. There is therefore a need for a common framework or language that will support a shared understanding of the concept of trust and permits the crafting of requirements of different stake-holders based on which trust gets developed across participating agents.

There is an urgent need of developing scalable TMMs for general acceptance of transactions ranging from micro-to-macro values over the Internet. There may be various levels of trust associated with the transaction; based on the value of transaction, which may depend on factors like - risk involved, information revealed during the process (privacy and confidentiality) etc. Further, TMMs should not overburden the user as that is counter to acceptability of trade. To reemphasize, it is imperative that the models be scalable as the volume of transactions over the Internet are invariably large.

Through *flexi*-ACL we have attempted to provide a flexible heterogeneous framework of TMM from which one could derive a TM system based upon the level of trust one demands. As highlighted already, *flexi*-ACL, supports hierarchical setups. Thus, it enables the specification of various contracts as discussed earlier. Implementation and performance analysis related to applications are in progress.

In general, trust has its own risks even in traditional commerce. For instance, a credit card user knows the risk in its usage; knowing the liability he will decide the places he will use. In the same way, it is very important to develop risk analysis techniques for TMMs that integrate various technological and non-technological parameters.

Search for a generic AC-cum-TMM for variety of applications in the field of security and distributed computing (e.g. IPSec, P2P, MANET, mobile agents, distributed file-systems, Grid etc., to name a few) is a very rich area for research. The urgency is very striking while deploying large applications with automated agents taking decisions on behalf of their owners. Programming such automation requires a framework on which applications can be built having capability to interact, negotiate and exchange credentials to reach an optimal contract agreeable to all involved parties. *flexi*-ACL is a step towards the larger goal of autonomic computing.

The work was done under a project supported by Department of Information Technology, Government of India. The authors thank the anonymous referees for useful suggestions. It is a pleasure to thank the editors of this special issue for their patience and support.

References

Al-Najjar N I, Casadesus-Masanell R 2001 Trust and discretion in agency contracts. In Strategy Unit Working Paper No. 02-06; HBS Working Paper No. 02-015. SSRN

Blaze M, Feigenbaum J, Strauss M 1998 Compliance checking in the PolicyMaker trust management system. *Financial cryptography*, LNCS 1465: 254–274

Blaze M, Feigenbaum J, Ioannidis J, Keromytis A 1999 The KeyNote Trust-management system. Version 2. RFC 2704, Internet Engineering Task Force

Chen R, Yeager W 2003 Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems. `http://www.jxta.org/docs/trust.pdf`

Christianson B, Harbison W S 1997 Why isn't trust transitive? In *Proc. Int. Workshop on Security Protocols* (London: Springer-Verlag) pp 171–176

Chu Y-H, Feigenbaum J, LaMacchia B, Resnick P, Strauss M 1997 REFEREE: Trust management for web applications. *World Wide Web J.* 2: 127–139

Clarke D, Elien J-E, Ellison C, Fredette M, Morcos A, Rivest R 2001 Certificate chain discovery in SPKI/SDSI. *J. Comput. Security* 9: 285–322

Ferraiolo D F, Sandhu R, Gavrila S, Kuhn D R, Chandramouli R 2000 A proposed standard for role-based access control. Technical report, National Institute of Standards & Technology

Ford W, Baum M S 2002 *Secure electronic commerce: Building the infrastructure for digital signatures and encryption* 2nd edn (Englewood Cliffs, NJ: Prentice Hall)

Hart O, Moore J 1999 Foundations of incomplete contracts. In *Review of economic studies* (Oxford: Blackwell) pp 115–138

Kamvar S D, Schlosser M T, Garcia-Molina H 2003 The EigenTrust algorithm for reputation management in P2P networks. In *Twelfth Int. World Wide Web Conference*

Kleinberg J 2000 The small-world phenomenon: An algorithmic perspective. In *32nd ACM Symp. on Theory of Computing (STOC)* (New York: ACM) pp 163–170

Levien R, Aiken A 2005 An attack-resistant, scalable name service. Draft submission to the Fourth International Conference on Financial Cryptography, http://www.levien.com/fc.ps

McDaniel P, Prakash A, Honeyman P 1999 Antigone: A flexible framework for secure group communication. In *8th USENIX UNIX Security Symposium* (San Franscisco: USENIX Assoc.) pp 99–114

Milgram S 1967 The small world problem. In *Psychol. Today* 1: 61–67

Miller M, Morningstar C, Frantz B 2001 Capability-based financial instruments. *Financial Cryptography* LNCS 1962: 349–378

Miller M, Shapiro J 2003 Paradigm regained: Abstraction mechanisms for access control. *Advances in computing science, ASIAN 2003 Programming Languages and Distributed Computation* LNCS 2896: 224–242

Page L, Brin S, Motwani R, Winograd T 1998 The PageRank citation ranking: Bringing order to the web. *7th Int. World Wide Web Conference* (WWW Consortium) pp 161–172

Patil V, Shyamasundar R K 2003a Notations for flexible access control system: *flexi*-ACL. Technical report, Tata Institute of Fundamental Research

Patil V, Shyamasundar R K 2003b ROADS: Role-based authorization and delegation system. *Int. Conf. on Computational & Experimental Engineering and Sciences*

Patil V, Shyamasundar R K 2004a Towards a flexible access control mechanism for e-transactions. In *ACM/IEEE Int. Workshop on Electronic Government, and Commerce: Design, Modeling, Analysis and Security (EGCDMAS-2004)* in conjunction with International Conference on E-Business and Telecommunication Networks (ICETE-2004) (Setubal, Portugal: INSTICC Press) pp 66–81

Patil V, Shyamasundar R K 2004b An efficient, secure and delegable micro-payment system. In *2004 IEEE Int. Conf. on e-Technology, e-Commerce and e-Service (EEE-04)* (New York: IEEE Comput. Soc.) pp 394–404

Stajano F, Anderson R 1999 The resurrecting duckling: Security issues for ad-hoc wireless networks. *7th Int. Workshop on Security Protocols*, LNCS (Heidelberg: Springer Verlag) 1796: 172–194

Xiong L, Liu L 2003 A reputation-based trust model for peer-to-peer ecommerce communities. In *IEEE Conf. on e-Commerce (CEC'03)* (New York: IEEE Comput. Soc.) pp 275–284

Ziegler C-N, Lausen G 2004 Spreading activation models for trust propagation. In *2004 IEEE Int. Conf. on e-Technology, e-Commerce and e-Service (EEE-04)* (New York: IEEE Comput. Soc.) pp 83–97

Zimmermann P 1995 *PGP source code and internals* (Cambridge, MA: MIT Press)